Procedimentos Graficos em Calculo Integral

Universidade Estadual de Maringá
Departamento de Matemática
Prof. Doherty Andrade (DMA- UEM)
Prof. Timothy M. (WLU-USA)

Maple

Este procedimento plota as seis regiões de integração expressas em coordenadas esféricas. As coordenadas esféricas são ρ (raio), θ (angulo longitudinal; i.e., ângulo feito por (x,y,0) e eixo x), e ϕ (ângulo latitudinal; i.e.,ângulo feito por (x,y,z) e o eixo positivo de z).

Este procedimento plota a fronteira da região no espaço quando descrita em coordendas esféricas (ρ, θ, ϕ), determinada pelas desigualdades

$\rho = f(\theta, \phi) .. g(\theta, \phi)$, $\theta = h(\phi) .. k(\phi)$, $\phi = a .. b$,

Isto é, ρ varia entre $f(\theta, \phi)$ e $g(\theta, \phi)$

e θ varia entre $h(\phi)$ e $k(\phi)$

e ϕ varia entre a e b .

Sintaxe: dpdtdphiplot($\rho = f(\theta, \phi) .. g(\theta, \phi)$, $\theta = h(\phi) .. k(\phi)$, $\phi = a .. b$,opts)

Os cinco procedimentos análogos são:

dpdphidtplot,

dtdpdphiplot,

dtdphidpplot,

dphidtdpplot,

dphidpdtplot,

todos com sintaxes análogas.

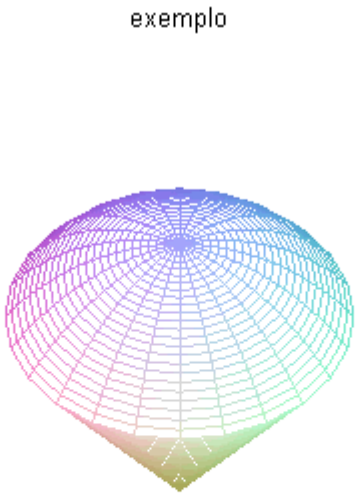
Execute o procedimento e faça os exemplos.

O Procedimento (execute-o)

Exemplos

Exemplos # dpdtdphiplot

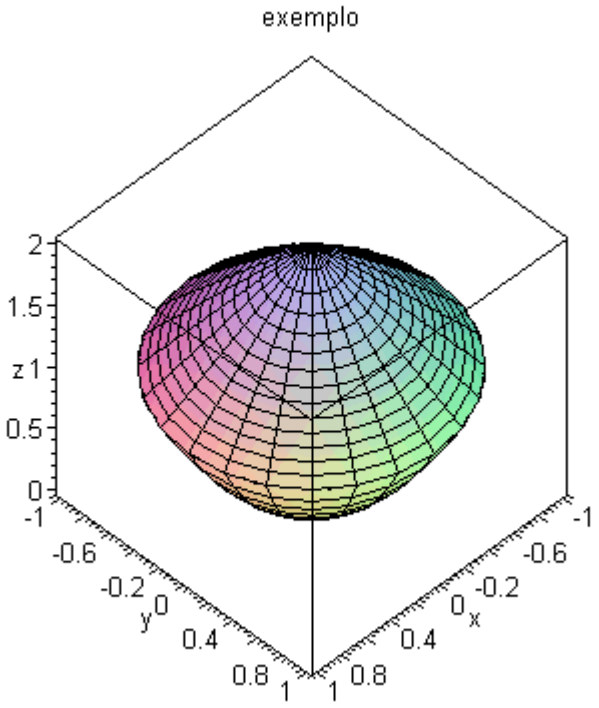
> **dpdphidtplot(rho=0..2,phi=0..Pi/4,theta=0..2*Pi, title=`exemplo`);**



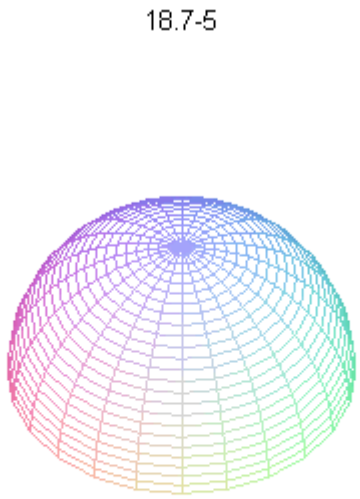
> **P1:=dpdtdphiplot(rho=sec(phi)..2/(cos(phi)+sin(phi)),theta=0..2*Pi,phi=0..Pi/4):**

> **P2:=dpdtdphiplot(rho=0..cot(phi)*csc(phi),theta=0..2*Pi,phi=Pi/4..Pi/2):**

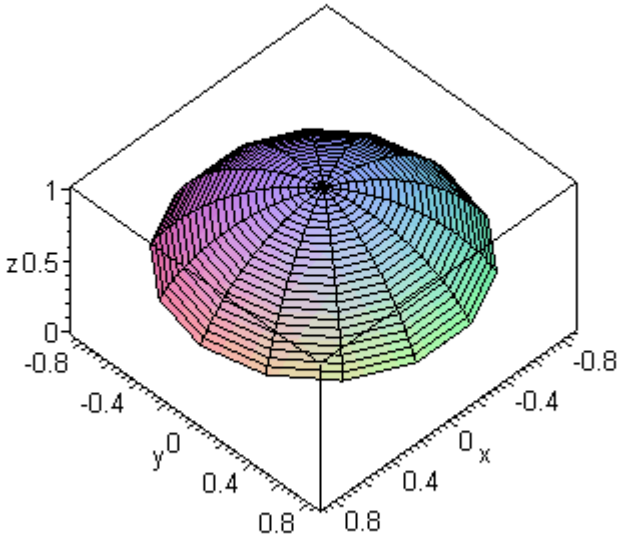
> **plots[display3d]({P1,P2},style=wireframe,title= `exemplo`);**



> **dpdphidtplot(rho=0..2*cos(phi),phi=0..Pi/4,theta=0..2*Pi, title= `18.7-5`);**

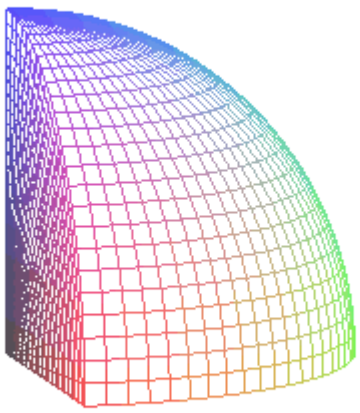


> **dphidpdtplot(phi=0..Pi/3,rho=0..1,theta=0..2*Pi);**



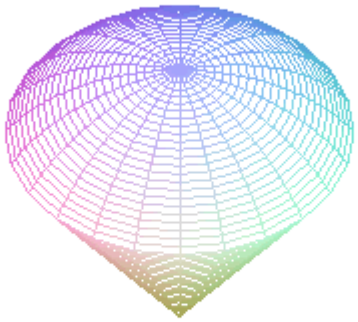
> **dphidpdtplot(phi=0..Pi/2,rho=0..1,theta=0..Pi/2,title= `esfera no primeiro octante`);**

esfera no primeiro octante



> **dpdphidtplot(rho=0..2*cos(phi),phi=0..Pi/6,theta=0..2*Pi, title=`cone de sorvete`);**

cone de sorvete



>

O Procedimento (execute-o)

#1. d(rho)d(theta)d(phi)

```
> dpdtdphiplot:= proc()
> local surface, f, f1, g, g1, h, h1, k, k1, a, b, p1, q1, t1, phi1, u,
> v, uloc, uloc1, uloc2, vloc, wloc, xloc, yloc, zloc, i, opt_seq,
> pbound, tbound, phibound;
> if nargs < 3 then
> ERROR(`there must be at least three arguments`) fi;
> pbound:=args[1];
> tbound:=args[2];
> phibound:=args[3];
> a:=op(1, rhs(phibound));
> b:=op(2, rhs(phibound));
> if not type (phibound, string = range) or not (op(1, pbound) = 'rho')or not (op(1, tbound) =
'theta') or not (op(1, phibound) = 'phi')
> or not type (evalf(a),numeric) or not type(evalf(b), numeric) then
> ERROR(`input expression is not of dpdtdphi type` ) fi;
> f:=op(1, rhs(pbound));
> g:=op(2, rhs(pbound));
> h:=op(1, rhs(tbound));
> k:=op(2, rhs(tbound));
> p1:=lhs(pbound);
> t1:=lhs(tbound);
> phi1:=lhs(phibound);
> f1:=unapply(f,(t1,phi1));
> g1:=unapply(g,(t1,phi1));
```

```
> h1:=unapply(h,phi1);
> k1:=unapply(k,phi1);
> uloc:=u*b+(1-u)*a;
> uloc1:=u*h1(a)+(1-u)*k1(a);
> uloc2:=u*h1(b)+(1-u)*k1(b);
> vloc:=v*k1(uloc)+(1-v)*h1(uloc);
> wloc:=v*f1(h1(uloc),uloc)+(1-v)*g1(h1(uloc),uloc);
> xloc:=v*f1(k1(uloc),uloc)+(1-v)*g1(k1(uloc),uloc);
> yloc:=v*f1(uloc1,a)+(1-v)*g1(uloc1,a);
> zloc:=v*f1(uloc2,b)+(1-v)*g1(uloc2,b);
> surface:={[f1(vloc,uloc),vloc,uloc],[g1(vloc,uloc),vloc,uloc],
> [wloc,h1(uloc),uloc], [xloc,k1(uloc),uloc],[yloc,uloc1,a],[zloc,uloc2,b]};
> if nargs =3 then
> q1:=plots[sphereplot](surface, u=0..1, v=0..1, axes=BOXED, grid
> =[15,25],style=PATCH, scaling=CONSTRAINED); fi;
> if nargs > 3 then
> opt_seq:=seq(args[i],`i`=4..nargs);
> q1:=plots[sphereplot](surface, u=0..1, v=0..1),opt_seq; fi;
> plots[display3d](q1);
> end:
#2. d(rho)d(phi)d(theta)
> dpdphidtplot:= proc()
> local surface, f, f1, g, g1, h, h1, k, k1, a, b, p1, q1, t1, phi1, u,
> v, uloc, uloc1, uloc2, vloc, wloc, xloc, yloc, zloc, i, opt_seq,
> pbound, tbound, phibound;
> if nargs < 3 then
```

```
> ERROR(`there must be at least three arguments`) fi;

> pbound:=args[1];

> phibound:=args[2];

> tbound:=args[3];

> a:=op(1, rhs(tbound));

> b:=op(2, rhs(tbound));

> if not type (phibound, string = range) or not (op(1, pbound) = 'rho')or not (op(1, tbound) =
'theta') or not (op(1, phibound) = 'phi')

> or not type (evalf(a),numeric) or not type(evalf(b), numeric) then

> ERROR(`input expression is not of dpdphidt type` ) fi;

> f:=op(1, rhs(pbound));

> g:=op(2, rhs(pbound));

> h:=op(1, rhs(phibound));

> k:=op(2, rhs(phibound));

> p1:=lhs(pbound);

> t1:=lhs(tbound);

> phi1:=lhs(phibound);

> f1:=unapply(f,(phi1,t1));

> g1:=unapply(g,(phi1,t1));

> h1:=unapply(h,t1);

> k1:=unapply(k,t1);

> uloc:=u*b+(1-u)*a;

> uloc1:=u*h1(a)+(1-u)*k1(a);

> uloc2:=u*h1(b)+(1-u)*k1(b);

> vloc:=v*k1(uloc)+(1-v)*h1(uloc);

> wloc:=v*f1(h1(uloc),uloc)+(1-v)*g1(h1(uloc),uloc);

> xloc:=v*f1(k1(uloc),uloc)+(1-v)*g1(k1(uloc),uloc);
```

```
> yloc:=v*f1(uloc1,a)+(1-v)*g1(uloc1,a);
> zloc:=v*f1(uloc2,b)+(1-v)*g1(uloc2,b);
> surface:={[f1(vloc,uloc),uloc,vloc],[g1(vloc,uloc),uloc,vloc],
> [wloc,uloc,h1(uloc)], [xloc,uloc,k1(uloc)],[yloc,a,uloc1],[zloc,b,uloc2]};
> if nargs =3 then
> q1:=plots[sphereplot](surface, u=0..1, v=0..1, axes=BOXED, grid
> =[15,25],style=PATCH, scaling=CONSTRAINED); fi;
> if nargs > 3 then
> opt_seq:=seq(args[i],`i`=4..nargs);
> q1:=plots[sphereplot](surface, u=0..1, v=0..1),opt_seq; fi;
> plots[display3d](q1);
> end:
#3. d(theta)d(phi)d(rho)
> dtdphidpplot:= proc()
> local surface, f, f1, g, g1, h, h1, k, k1, a, b, p1, q1, t1, phi1, u,
> v, uloc, uloc1, uloc2, vloc, wloc, xloc, yloc, zloc, i, opt_seq,
> pbound, tbound, phibound;
> if nargs < 3 then
> ERROR(`there must be at least three arguments`) fi;
> tbound:=args[1];
> phibound:=args[2];
> pbound:=args[3];
> a:=op(1, rhs(pbound));
> b:=op(2, rhs(pbound));
> if not type (phibound, string = range) or not (op(1, pbound) = 'rho')or not (op(1, tbound) =
'theta') or not (op(1, phibound) = 'phi')
> or not type (evalf(a),numeric) or not type(evalf(b), numeric) then
```

```
> ERROR(`input expression is not of dtdphidp type` ) fi;
> f:=op(1, rhs(tbound));
> g:=op(2, rhs(tbound));
> h:=op(1, rhs(phibound));
> k:=op(2, rhs(phibound));
> p1:=lhs(pbound);
> t1:=lhs(tbound);
> phi1:=lhs(phibound);
> f1:=unapply(f,(phi1,p1));
> g1:=unapply(g,(phi1,p1));
> h1:=unapply(h,p1);
> k1:=unapply(k,p1);
> uloc:=u*b+(1-u)*a;
> uloc1:=u*h1(a)+(1-u)*k1(a);
> uloc2:=u*h1(b)+(1-u)*k1(b);
> vloc:=v*k1(uloc)+(1-v)*h1(uloc);
> wloc:=v*f1(h1(uloc),uloc)+(1-v)*g1(h1(uloc),uloc);
> xloc:=v*f1(k1(uloc),uloc)+(1-v)*g1(k1(uloc),uloc);
> yloc:=v*f1(uloc1,a)+(1-v)*g1(uloc1,a);
> zloc:=v*f1(uloc2,b)+(1-v)*g1(uloc2,b);
> surface:={[uloc,f1(vloc,uloc),vloc],[uloc,g1(vloc,uloc),vloc],
> [uloc,wloc,h1(uloc)], [uloc,xloc,k1(uloc)],[a,yloc,uloc1],[b,zloc,uloc2]};
> if nargs =3 then
> q1:=plots[sphereplot](surface, u=0..1, v=0..1, axes=BOXED, grid
> =[15,25],style=PATCH, scaling=CONSTRAINED); fi;
> if nargs > 3 then
```

```
> opt_seq:=seq(args[i],`i`=4..nargs);
> q1:=plots[sphereplot](surface, u=0..1, v=0..1),opt_seq; fi;
> plots[display3d](q1);
> end:
#4. d(phi)d(theta)d(rho)
> dphidtdpplot:= proc()
> local surface, f, f1, g, g1, h, h1, k, k1, a, b, p1, q1, t1, phi1, u,
> v, uloc, uloc1, uloc2, vloc, wloc, xloc, yloc, zloc, i, opt_seq,
> pbound, tbound, phibound;
> if nargs < 3 then
> ERROR(`there must be at least three arguments`) fi;
> phibound:=args[1];
> tbound:=args[2];
> pbound:=args[3];
> a:=op(1, rhs(pbound));
> b:=op(2, rhs(pbound));
> if not type (phibound, string = range) or not (op(1, pbound) = 'rho')or not (op(1, tbound) =
'theta') or not (op(1, phibound) = 'phi')
> or not type (evalf(a),numeric) or not type(evalf(b), numeric) then
> ERROR(`input expression is not of dphidtdp type` ) fi;
> f:=op(1, rhs(phibound));
> g:=op(2, rhs(phibound));
> h:=op(1, rhs(tbound));
> k:=op(2, rhs(tbound));
> p1:=lhs(pbound);
> t1:=lhs(tbound);
> phi1:=lhs(phibound);
```

```
> f1:=unapply(f,(t1,p1));
> g1:=unapply(g,(t1,p1));
> h1:=unapply(h,p1);
> k1:=unapply(k,p1);
> uloc:=u*b+(1-u)*a;
> uloc1:=u*h1(a)+(1-u)*k1(a);
> uloc2:=u*h1(b)+(1-u)*k1(b);
> vloc:=v*k1(uloc)+(1-v)*h1(uloc);
> wloc:=v*f1(h1(uloc),uloc)+(1-v)*g1(h1(uloc),uloc);
> xloc:=v*f1(k1(uloc),uloc)+(1-v)*g1(k1(uloc),uloc);
> yloc:=v*f1(uloc1,a)+(1-v)*g1(uloc1,a);
> zloc:=v*f1(uloc2,b)+(1-v)*g1(uloc2,b);
> surface:={[uloc,vloc,f1(vloc,uloc)],[uloc,vloc,g1(vloc,uloc)],
> [uloc,h1(uloc),wloc], [uloc,k1(uloc),xloc],[a,uloc1,yloc],[b,uloc2,zloc]};
> if nargs =3 then
> q1:=plots[sphereplot](surface, u=0..1, v=0..1, axes=BOXED, grid
> =[15,25],style=PATCH, scaling=CONSTRAINED); fi;
> if nargs > 3 then
> opt_seq:=seq(args[i],`i`=4..nargs);
> q1:=plots[sphereplot](surface, u=0..1, v=0..1),opt_seq; fi;
> plots[display3d](q1);
> end:
#5. d(theta)d(rho)d(phi)
> dtdpdphiplot:= proc()
> local surface, f, f1, g, g1, h, h1, k, k1, a, b, p1, q1, t1, phi1, u,
> v, uloc, uloc1, uloc2, vloc, wloc, xloc, yloc, zloc, i, opt_seq,
```

```
> pbound, tbound, phibound;

> if nargs < 3 then

> ERROR(`there must be at least three arguments`) fi;

> tbound:=args[1];

> pbound:=args[2];

> phibound:=args[3];

> a:=op(1, rhs(pbound));

> b:=op(2, rhs(pbound));

> if not type (phibound, string = range) or not (op(1, pbound) = 'rho')or not (op(1, tbound) =
'theta') or not (op(1, phibound) = 'phi')

> or not type (evalf(a),numeric) or not type(evalf(b), numeric) then

> ERROR(`input expression is not of dtdpdphi type` ) fi;

> f:=op(1, rhs(tbound));

> g:=op(2, rhs(tbound));

> h:=op(1, rhs(phibound));

> k:=op(2, rhs(phibound));

> p1:=lhs(pbound);

> t1:=lhs(tbound);

> phi1:=lhs(phibound);

> f1:=unapply(f,(p1,phi1));

> g1:=unapply(g,(p1,phi1));

> h1:=unapply(h,p1);

> k1:=unapply(k,p1);

> uloc:=u*b+(1-u)*a;

> uloc1:=u*h1(a)+(1-u)*k1(a);

> uloc2:=u*h1(b)+(1-u)*k1(b);

> vloc:=v*k1(uloc)+(1-v)*h1(uloc);
```

```
> wloc:=v*f1(h1(uloc),uloc)+(1-v)*g1(h1(uloc),uloc);
> xloc:=v*f1(k1(uloc),uloc)+(1-v)*g1(k1(uloc),uloc);
> yloc:=v*f1(uloc1,a)+(1-v)*g1(uloc1,a);
> zloc:=v*f1(uloc2,b)+(1-v)*g1(uloc2,b);
> surface:={[vloc,f1(vloc,uloc),uloc],[vloc,g1(vloc,uloc),uloc],
> [h1(uloc),wloc,uloc], [k1(uloc),xloc,uloc],[uloc1,yloc,a],[uloc2,zloc,b]};
> if nargs =3 then
> q1:=plots[sphereplot](surface, u=0..1, v=0..1, axes=BOXED, grid
> =[15,25],style=PATCH, scaling=CONSTRAINED); fi;
> if nargs > 3 then
> opt_seq:=seq(args[i],`i`=4..nargs);
> q1:=plots[sphereplot](surface, u=0..1, v=0..1),opt_seq; fi;
> plots[display3d](q1);
> end:
#6. d(phi)d(rho)d(theta)
> dphidpdtplot:= proc()
> local surface, f, f1, g, g1, h, h1, k, k1, a, b, p1, q1, t1, phi1, u,
> v, uloc, uloc1, uloc2, vloc, wloc, xloc, yloc, zloc, i, opt_seq,
> pbound, tbound, phibound;
> if nargs < 3 then
> ERROR(`there must be at least three arguments`) fi;
> phibound:=args[1];
> pbound:=args[2];
> tbound:=args[3];
> a:=op(1, rhs(tbound));
> b:=op(2, rhs(tbound));
```

```
> if not type (phibound, string = range) or not (op(1, pbound) = 'rho')or not (op(1, tbound) = 'theta') or not (op(1, phibound) = 'phi')

> or not type (evalf(a),numeric) or not type(evalf(b), numeric) then

> ERROR(`input expression is not of dphidpdt type` ) fi;

> f:=op(1, rhs(phibound));

> g:=op(2, rhs(phibound));

> h:=op(1, rhs(pbound));

> k:=op(2, rhs(pbound));

> p1:=lhs(pbound);

> t1:=lhs(tbound);

> phi1:=lhs(phibound);

> f1:=unapply(f,(p1,t1));

> g1:=unapply(g,(p1,t1));

> h1:=unapply(h,t1);

> k1:=unapply(k,t1);

> uloc:=u*b+(1-u)*a;

> uloc1:=u*h1(a)+(1-u)*k1(a);

> uloc2:=u*h1(b)+(1-u)*k1(b);

> vloc:=v*k1(uloc)+(1-v)*h1(uloc);

> wloc:=v*f1(h1(uloc),uloc)+(1-v)*g1(h1(uloc),uloc);

> xloc:=v*f1(k1(uloc),uloc)+(1-v)*g1(k1(uloc),uloc);

> yloc:=v*f1(uloc1,a)+(1-v)*g1(uloc1,a);

> zloc:=v*f1(uloc2,b)+(1-v)*g1(uloc2,b);

> surface:={[vloc,uloc,f1(vloc,uloc)],[vloc,uloc,g1(vloc,uloc)],

> [h1(uloc),uloc,wloc], [k1(uloc),uloc,xloc],[uloc1,a,yloc],[uloc2,b,zloc]};

> if nargs =3 then

> q1:=plots[sphereplot](surface, u=0..1, v=0..1, axes=BOXED, grid
```

```
> =[15,25],style=PATCH, scaling=CONSTRAINED); fi;
> if nargs > 3 then
> opt_seq:=seq(args[i],`i`=4..nargs);
> q1:=plots[sphereplot](surface, u=0..1, v=0..1),opt_seq; fi;
> plots[display3d](q1);
> end:
```